What is claimed is:

1. A method for disabling clocks to at least one processor core of a plurality of processor cores comprising:

calculating an executing core limit based at least in part on a workload;

executing an n number of available threads, wherein n is an integer,

enabling an m number of processor cores, wherein m is an integer and is less than or equal to n, the number of available threads.

2. The method of claim 1 wherein disabling clocks to at least one processor during an idle time period as the processor core waits for a memory operation.

3. The method of claim 1 wherein disabling clocks to at least one processor core results in decreased power consumption.

4. The method of claim 1 wherein disabling clocks to at least one processor core allows for increasing the operating frequency of that processor core.

5. A method for selecting a voltage and frequency operating point to at least one processor core of a plurality of processor cores comprising:

predicting an activity level of a plurality of threads running on all of the plurality of processor cores;

enabling a subset of the plurality of processor cores based at least in part on the

activity level.

6. The method of claim 1 wherein the activity level is an executing core limit that is based at least in part on adhering to thermal power considerations.

[0029] 7. The method of claim 6 wherein the executing core limit is based at least in part on a formula, wherein N depicts the number of threads that have context; %E depicts the percentage executing time; and %M depicts the percentage memory reference time. and the formula is :

[0030] int (N x (%E/ (%E + %M)))

8. A state diagram for a plurality of multi-core processors comprising:

A first state for a core without an assigned thread;

A second state for a queue to store cores with an assigned thread;

A third state for enabling the core to run a thread; and

A fourth state to disable a core.

9. The state diagram of claim 8 wherein the queue is a first in first out (FIFO) queue.

10. The state diagram of claim 8 wherein the core transitions from a second state to the third state if the number of enabled cores is less than an executing core limit.

[0031] 11. The state diagram of claim 10 wherein the executing core limit is based at least in part on a formula, wherein N depicts the number of threads that have context; %E depicts the percentage executing time; and %M depicts the percentage memory reference time. and the formula is :

[0032] int (N x (%E/ (%E + %M)))

12. The state diagram of claim 8 wherein the core transitions from a third state to the fourth state if the core is idle as it waits for completion of a memory operation.

13. A method for a state diagram for a plurality of multi-core processors comprising:

assigning a first state to a core without an assigned thread;

assigning a second state for a queue to store cores with an assigned thread;

comparing the number of enabled cores to an executing core limit, assigning a third state for enabling the core to run a thread if the number of enabled cores is less than the executing core limit; and

assigning a fourth state to disable a core.

14. The method of claim 13 wherein the queue is a first in first out (FIFO) queue.

[0033] 15. The method of claim 13 wherein the executing core limit is based at least in part on a formula, wherein N depicts the number of threads that have context; %E depicts the percentage executing time; and %M depicts the percentage memory reference time. and the formula is :

[0034] int (N x (%E/ (%E + %M)))

16. The state diagram of claim 13 wherein the core transitions from a third state to the fourth state if the core is idle as it waits for completion of a memory operation.

17. A system of multi-core processors comprising:

at least one multi-core processor coupled to a cache memory, and coupled to at least two clockwise directional busses to receive requests and responses; and

a core rationing logic to manage the number of enabled cores to be less than or equal to an executing core limit.

[0035] 18. The system of claim 17 wherein the executing core limit is based at least in part

on a formula, wherein N depicts the number of threads that have context; %E depicts the percentage executing time; and %M depicts the percentage memory reference time. and the formula is :

[0036] int (N x (%E/ (%E + %M))).

19. The system of claim 17 further comprises a system interface that contains:

a plurality of memory controllers for memory DIMMs;

a router logic to handle the interconnection links to other processor dies or I/O subsystems; and

the core rationing logic.

20. The system of claim 17 further comprises at least two counter- clockwise directional busses to receive requests and responses.

21. The system of claim 17 wherein the cache memory is a level three (L3) memory with a plurality of independent memory banks.